Convert Photo to PDF

```
from PIL import Image
import os

def convert_images_to_pdf(input_files, output_pdf):
    images = []

    # If a single file is provided as input
    if isinstance(input_files, str):
        input_files = [input_files]

    for file_path in input_files:
        try:
            img = Image.open(file_path)
            # Convert the image to RGB mode for PDF compatibility
            if img.mode != 'RGB':
                img = img.convert('RGB')
            images.append(img)
        except Exception as e:
            print(f"Error processing {file_path}: {e}")

    if not images:
        print("No valid images to convert.")
        return

    # Save as single PDF
    images[0].save(output_pdf, save_all=True,
append_images=images[1:])
    print(f"PDF created successfully: {output_pdf}")

# Example usage
input_files = [
    "path\file.jpg",
    "path\file.png"  # Add more file paths here if needed
]

output_pdf_path = "path\file.pdf"

# Check if it's a single file or folder
if len(input_files) == 1:
    convert_images_to_pdf(input_files[0], output_pdf_path)
else:
    convert_images_to_pdf(input_files, output_pdf_path)
```

Convert pptx to PDF

```
import os
from comtypes import client
from PyPDF2 import PdfMerger

def convert_ppt_to_pdf(input_files):
    if isinstance(input_files, str):
        input_files = [input_files]

    powerpoint = client.CreateObject("Powerpoint.Application")
    powerpoint.WindowState = 2  # Minimize PowerPoint

    pdf_paths = []

    for input_file in input_files:
        try:
            # Temporary PDF path in the same directory as the input
file
            pdf_path = os.path.splitext(input_file)[0] + ".pdf"

            presentation = powerpoint.Presentations.Open(input_file,
WithWindow=False)
            presentation.SaveAs(pdf_path, 32)  # Save as PDF
            presentation.Close()

            print(f"Converted {input_file} to {pdf_path}")
            pdf_paths.append(pdf_path)
        except Exception as e:
            print(f"Error converting {input_file}: {e}")

    powerpoint.Quit()
    return pdf_paths

def merge_pdfs(pdf_paths, merged_output_pdf):
    merger = PdfMerger()
    try:
        for pdf_path in pdf_paths:
            merger.append(pdf_path)
        merger.write(merged_output_pdf)
        merger.close()
        print(f"PDFs merged successfully into: {merged_output_pdf}")
    except Exception as e:
        print(f"Error merging PDFs: {e}")

def main():
    input_files = [
        r"path\file.ppt",
        r"path\file.pptx"
    ]
```

```
    merged_output_pdf_path = "path\file.pdf"

    # Convert presentations to individual PDFs
    pdf_paths = convert_ppt_to_pdf(input_files)

    # Merge PDFs into one file
    merge_pdfs(pdf_paths, merged_output_pdf_path)

    # Cleanup temporary PDF files
    for pdf_path in pdf_paths:
        os.remove(pdf_path)
        print(f"Deleted temporary file: {pdf_path}")

if __name__ == "__main__":
    main()

from PyPDF2 import PdfMerger

def merge_pdfs(pdf_files, output_pdf):
    merger = PdfMerger()
    try:
        for pdf_file in pdf_files:
            merger.append(pdf_file)
        merger.write(output_pdf)
        merger.close()
        print(f"PDFs merged successfully into: {output_pdf}")
    except Exception as e:
        print(f"Error merging PDFs: {e}")

# Example usage
pdf_files = [
    r"path\file.pdf",
    r"path\file.pdf"
]
output_pdf_path = "path\file.pdf"

merge_pdfs(pdf_files, output_pdf_path)
```

THE END @:-)
